Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Procedia Computer Science 21 (2013) 60 – 67
Procedia Computer Science

The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013)

OBAME: Optimized Bio-inspired Algorithm to Maximize Search Efficiency in P2P Databases

Heba A. Kurid[a*], Thanaa S. Alnusairi[b], Hajar S. Almujahed[a]

[a]*Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia*
[b]*Al Jouf University, Al-Jawf, Saudi Arabia*

Abstract

P2P databases are characterized by high site-failure rates, unpredictable network topology and complex management, due to the complete absence of a centralized controller. These characteristics have introduced novel challenges and research issues to the field. Among the most difficult challenges is the process of locating information among various participants in the network. This paper presents an original contribution by proposing an Optimized Bio-inspired Algorithm to maximize search efficiency in P2P databases (OBAME). Experimental results showed that OBAME outperformed Ant- and Bee- Inspired algorithms in terms of network traffic and query response time.

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: P2P Database; Centralized Peer-to-Peer Network; Ant Colony Optimization; Bee Colony Optimization, Bio-inspired Algorithms

1. Introduction

A Peer-to-Peer (P2P) database system consists of multiple, autonomous, component databases, connected via a network, that interact and share information and resources to achieve mutual common goals. This configuration offers ubiquitous access to a shared pool of resources that can be rapidly provisioned, eliminating the need for centralized servers to store data and manage resources across databases.

On the other hand, P2P databases are characterized by high site-failure rates, an unpredictable network topology and complex management, due to the complete absence of a centralized controller. These

* Corresponding author. Tel.: +966 555 669791
E-mail address: hakurdi@imamu.edu.sa.

characteristics have introduced novel challenges and research issues to the field. Among the most difficult challenges is the process of locating information among various participants in the network. This paper presents an original contribution by proposing an Optimized Bio-inspired Algorithm to maximize search Efficiency in P2P databases (OBAME).

Bio-inspired algorithms are increasingly attracting attention in solving optimization problems. In [1] an overview of biological facts about social insects, their inspired algorithms and application areas in computer engineering and science are presented. Among all social insects, the foraging behavior of ant and bee colonies, and how it can be used to solve search problems, is particularly popular.

However, all of the previous works have considered only a single specific insect colony, either ant or bee. Observing that each colony shows a more successful behavior in dealing with particular situations, this paper proposed a novel adaptive approach that combines the behavior of both colonies. Therefore, search agents would work as forager ants where conditions favor that behavior, or work as forager bees, if the conditions suit the latter. The main objective was to optimize the query search process by minimizing the search area. This has reflected positively in minimizing search query response time and reducing network traffic.

This paper is organized as follows: Section 2 provides an overview of the related work. The OBAME design, biological inspiration and workflow are illustrated in Section 3. Section 4 highlight the implementation tools and environments while experimental results and discussion are presented in Section 5. Finally, the paper is concluded in Section 6 indicating some directions for future work.

2. Related Work

The Gnutella system [2] is the most known paradigm of P2P systems. Its search technique is implemented based on a simple flooding strategy that does not take the network traffic into considerations. When a node wants to initiate a query, it sends the query to all of its neighbors, each of which forwards the query to all of its neighbors, and so on, until the maximum number of hops is reached or the time-to-live (TTL) equals zero.

DiCAS [3] has been proposed to enhance the Gnutella search by logically organising all peers into multiple layers. Each layer holds all peers with the same group ID. The query is flooded only to the layer, with the matching group ID, rather than being flooded to all peers in the entire network. Although this has considerably enhanced the congestion problem resulted from the Gnutella pure flooding strategy, the query searching time and network traffic is still high.

Due to the increasing demand to enhance the search efficiency in P2P databases, various Artificial Intelligence (AI) approaches, specifically Swarm Intelligence (SI), have been utilized. The bio- inspired optimization is usually based on intelligent foraging behavior of social insects such as ants and bees. Ant Colony Optimization (ACO) [4] and Bee Colony Optimization (BCO) [5] are well established algorithms employed to guide a search process to the best route to information sources.

Basically, Ant Colony-inspired Algorithms (ACAs) utilize ideas from the natural ant colonies where basically, a group of forager ants starts searching for food sources randomly. Once an ant found a source, it evaluates its profitability then returns back to the nest. During her return trip, the ant deposits an amount of pheromone proportional to the profitability of the discovered source [6]. So, when other ants search for food sources, they would select the route with the highest pheromone level. ACAs in P2P databases optimize the query search process to be selective rather than flooding the queries to all peer databases. Thus, the route to the database with the greater possibility of success in getting the result, i.e. the promising database, is assigned the highest promising value. As a result, more queries will be directed to it [7].

On the other hand, Bee Colony-inspired Algorithms (BCAs) utilize ideas from the natural bee colonies where a group of forager bees starts searching randomly for food sources. Once a bee found one, it assess the profitability of the source based on the amount of the available food. If the profitability is high, the bee returns to the hive and advertises the source by dancing at the entrance of the hive, in an area called the dancing floor. Bees who have not already known a nectar source, or their last visited resources have low profitability, would randomly select a dancing bee and fly to its advertised resource. Other bees would fly directly to their already known sources [8]. BCAs can enhance the searching process in a file sharing P2P database by minimizing the search area to a group of promising databases which contains databases with higher promising values [9]. In an investigation of whether ACAs are outperformed by BCAs, experimental results by [10] showed that BCAs are significantly faster when finding and collecting food and use fewer time steps to complete a task.

Based on the above, it is clear that although an ACA would insure directing all ants to the highest profitable source, it suffers from some problems when compared to BCAs. First, information about a good food source is not directly advertised. It takes time until the pheromone reaches a certain concentration, so ants can get to know about this source. On the other hand, in a bee colony a good food source is immediately advertised to the entire colony. Second, to make a decision, an ant needs to compare several alternatives which is time consuming and requires information about other alternatives from outside. This is not the case in bees where the decision is local to the bee itself requiring no outside information. Third, forager ants would always be directed to the resource with the highest profitability causing congestion and consuming bandwidth. While in bee colony, forager bee will be distributed randomly among a group of resource with profitability value above a certain threshold. Therefore, OBAME has been designed with an adaptive strategy based on ACAs and BCAs to take advantages of both groups of algorithm to optimize the query routing strategy leading to more efficient search in P2P databases.

3. Proposed Algorithm (OBAME)

OBAME goal is to optimise the query search by minimizing the search area. This should be refelected positively in network traffic and query response time. With this goal in mind, the algorithm infrastructure, its components and workflow were designed.

3.1. System Architecture

The assumed infrastructure of OBAME P2P system is a P2P network with an indexing server. Napster [11] is a well established paradigm that follows this architecture where a file sharing service uses an index to manage all file links shared by peers. Peers register shared data information on the index server. The address information of file owners is given to the requester from the index server. Then the file is transferred between the requester and the owner directly. Based on this, OBAME system is composed of the following components:

- Indexing Server: It is the central server running OBAME and controlling routing queries over connected peers. It contains the information about the best routes for all previously searched queries.
- Peer Nodes: They are the connected peers with a local database for each. They have a common goal of sharing and exchanging information. Each peer node is composed of:
  - User Interface: it maintains the connection between the user and the entire system. The user can enter its input and receive the result through it.
  - Query Manager: manages the queries that may run locally on the local database (LDB) or globally over the P2PDBS through the indexing server.
  - Local Database Management System (LDBMS): it is the local database management system, which is responsible for managing the local database of the peer node.
  - P2P Manager: It manages the connection between the peers. It is responsible of establishing the connection between each peer and the peers in the system.

3.2. Biological Inspiration

Table 1, illustrates the biological inspiration of OBAME by indicating the correspondence between related terms in OBAME, ACAs and BCAs.

Table 1, Biological inspiration of OBAME

OBAME	ACA	BCA
Promising database	Food source	Food source
Promising value	Profitability	Profitability
Server	Nest	Hive
Routing table	Explored routs	Dancing floor
Search Agent	Forager ant	Forager bee

3.3. System Workflow

The OBAME workflow is illustrated by the flowchart in Fig. 1. Initially, a user, on the peer side, enters a search query. The query is received by the indexing server which checks the routing table if this query has been already searched for. If the routing table does not contain information related to the query, the server creates searching agents to start searching randomly for promising databases. When an agent finds a related result, it calculates the promising value of the database and sends the value to the server. The server updates the routing table accordingly.

On the other hand, if there are already exist promising values related to the entered search query, in the routing table, then the server randomly chooses a peer databases with a promising value greater than a threshold. It is important to note that since the server chooses a promising database randomly, not all peers follow the same path which significantly reduces the congestion in network [8]. When there is not a promising value higher than the threshold, the server selects the peer database with the highest promising value from all related promising databases.

Finally, the server sends the routing information related to the promising database to the requesting peer so it can send its search query directly to that peer with the promising database. The OBAME search function is shown in Fig. 2.

4. Implementation

The P2P network was implemented as 20 nodes representing medical research centers that share their research studies in relational databases. Each node has a local database that contains tables related to its treated diseases information which includes: disease name, symptoms, laboratory analysis results, treatments and finally names of researchers who studied those diseases. Databases are considered homogeneous and capable of handling various tasks such as searching, creating, modifying and deleting databases, tables, fields and rows.

NetBeans IDE 7.3, JDK 7 and wampServer:Mysql were used to implement the overall system. Nodes were implemented in php and connected to databases using JDBC-ODBC bridge. Communication among the server and nodes was implemented based on Java RMI.

Initially, OBAME bootstraps the system by searching all of the connected 20 nodes for a set of pre-defined search queries. Promising values, for each query are calculated and stored as initial values in the routing table, within the server.

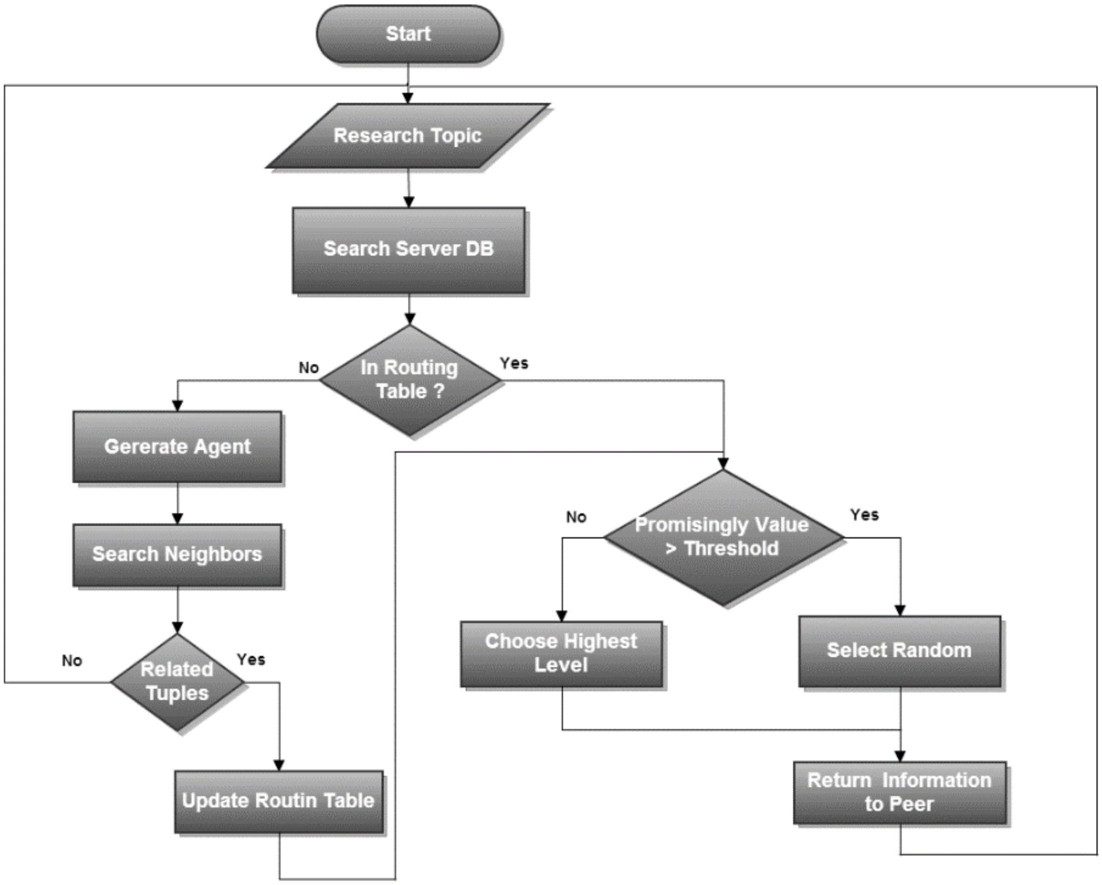


Fig. 1: OBAME Workflow.

```
Int OBMEsearch (searchKeyword)
// the input: search Keyword
// returns the peerID (int) which contains the promising database or print an error message then exits
1.    Check routingTable;
2.    if (routingTable does not contain information related to the searchKeyword)
         2.1 Create agents to search randomly;   //act as both  forager bees and ants
         2.2 if  (match found for searchKeyword)
                  2.2.1 Calculate promisingValue
                    2.2.1 Update routingTable
3.    else
         3.1 Print "No matching results";
         3.2 Exit;
4.    if (promisingValue > threshold)                 //act as bee
          3.1 peerID = Choose prmisingDatabase randomly;
5.    else                                            //act as ant
       4.1 peerID = Choose prmisingDatabase with the highest promisingValue;
6.    return peerID ;
7.    End
```

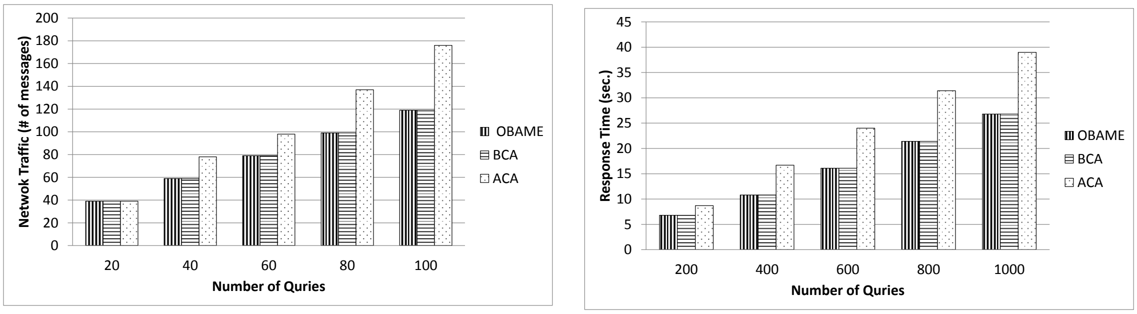
Fig. 2: OBAME Search Algorithm

5. Results and Discussion

An experimental framework to evaluate OBAME was designed with the objective of assessing the performance of the algorithm by measuring the network traffic in promising databases and the search query response time. An ACA and a BCA were used as benchmark algorithms. As there are some cases in which ACAs perform better than BCAs and vice versa, OBAME has taken distinct features from each. Two experiments were perform: first when the query results in promising databases are above the threshold, second when the query results are less than the threshold.

5.1. Query results in promising databases are above the threshold

Fig. 3(a) shows how OBAME, BCA and ACA affects the network traffic, of the 20 nodes under various number of queries [20, 40, 60, 80, 100], when the query results in promising databases are above the threshold. The figure indicates the growth tendency of the number of the travelling messages in the network in general as the number of queries increases. However, it has higher values in ACA than OBAME and BCA. This is because in ACA, all traffic is sent to the database with the highest promising value. In contrast, OBAME and BCA have identical performance; OBAME search agents would adopt the behavior of forager bees. So, selecting a database from the set of promising databases will be done randomly, reducing the traffic and evenly distributing it between nodes.



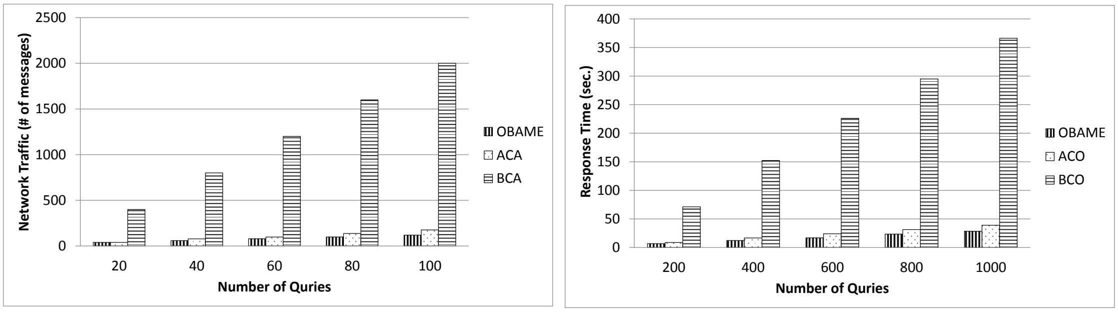
(a) Network traffic (b) Query response time

Fig. 3: Performance of OBAME, BCA and ACA when Query results are above the threshold

Fig. 3(b) shows how OBAME, BCA and ACA affects the query response time of the 20 nodes under various number of queries [200, 400, 600, 800, 1000] when the query results in promising databases are above the threshold. The x-axis points to number of queries for a given topic and y-axis points to response time (in seconds) of queries. Basically, it was expected that as OBAME adopts BCA strategy that chooses randomly between promising databases, both algorithms would show identical superior results when compared to ACA. To test this idea, 200 queries were sent searching for the same topic. Each time OBAME and BCA choose randomly between promising databases. In contrast, ACA chooses the highest one. The same results were obtained as the number of queries increased gradually to 1000 quires.

5.2. Query results in promising databases are below the threshold

Fig. 4(a) illustrates the performance of OBAME, BCA and ACA under various number of queries [20, 40, 60, 80, 100] in terms of the amount of network traffic when the query results, in promising databases, are below the threshold. In this case, BCA showed the worst performance among the three algorithms. This is because when a forager bee finds a food source (promising database for a certain query) and notices that its profitability (promising value) less than the threshold, it does not advertise the source. Thus, each time this topic is requested, the agents in BCA start searching randomly, dramatically increasing the network traffic, unless the requesting peer has requested this topic before where the query route was stored locally. On the other hand, OBAME would choose the highest promisingly value (works as ACA) and routing the query directly to it.



(a) Network traffic (b) Query response time

Fig. 4: Performance of OBAME, BCA and ACA when Query results are below the threshold

Fig. 4(b) shows how OBAME, BCA and ACA affects the query response time of the 20 nodes under various number of queries [200, 400, 600, 800, 1000] when the query results in promising databases are below the threshold. It is apparent that BCA has the worst performance in this case also due to the same reasons explained above. On the other hand, OBAME, has the best performance working based on ACA strategy and choosing the database with the highest promising value shortening the response time.

6. Conclusion

This paper proposed a novel optimizing algorithm for better the routing strategies of queries in P2P databases. The algorithm combines techniques from ant colony-inspired algorithms (ACAs) and bee colony-inspired algorithms (BCAs), to eliminate flooding and a directly route the query to the node that contains the promising data base. Experimental results showed that OBAME has successfully outperformed both algorithms by decreasing network traffic and response time in all scenarios.

For future work it is suggested to augment OBAME with the semantic techniques, so quires are routed to the promising database that has the most semantically matching search keyword.

References

[1] A. Abraham, C. Grosan, V. Ramos, Eds. *Stigmergic Optimization.* Springer, 2006.

[2] Miller, Jeffrey. "Characterization of data on the gnutella peer-to-peer network." In proceeding of the *First IEEE Consumer Communications and Networking Conference, CCNC 2004.*, 2004, pp. 489-494.

[3] Wang, Chen, Li Xiao, Yunhao Liu, and Pei Zheng. "DiCAS: An efficient distributed caching mechanism for P2P systems." In proceeding *Parallel and Distributed Systems, IEEE Transactions on* 17, 2006, pp. 1097-1109.

[4] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." In proceeding *Computational Intelligence Magazine, IEEE* 1, 2006. pp. 28-39.

[5] Wedde, Horst F., Muddassar Farooq, and Yue Zhang. "BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior." In proceeding *Ant colony optimization and swarm intelligence*, Springer Berlin Heidelberg, 2004, pp. 83-94.

[6] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni. "Ant system: optimization by a colony of cooperating agents." In proceeding *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 26, 1996, pp. 29-41.

[7] Valêncio, Carlos Roberto, Leandro Rincon Costa, C. H. El Hetti Laurenti, and Adriano Mauro Cansian. "Application of Ontologies in Peer-to-Peer Database with Routing Queries Based on Ant Colony Algorithm." In proceeding *Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2011 12th International Conference on, pp. 287-292. IEEE, 2011.

[8] Kurdi, H., M. Li, and H. S. Al-Raweshidy. "A Generic Framework for Resource Scheduling in Personal Mobile Grids Based on Honeybee Colony." In proceeding *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on*, pp. 297-302. IEEE, 2008.

[9] K. Dhurandher, S. Misra, P. Pruthi, S. Singhal, S. Aggarwal and I. Woungang, "A Swarm Intelligence-based P2P file sharing protocol using Bee Algorithm," Proc. of the 7th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-09), May 10-13, 2009.

[10] N.P. Lemmens, "To bee or not to bee: A comparative study in swarm intelligence," M.S. Thesis, Maastricht University, Maastricht ICT Competence Centre, Institute for Knowledge and Agent Technology, Maastricht, Netherlands, 2006.

[11] B., Lee, "Reflections On Napster: The Ninth Circuit Takes A Walk On The Wild Side". *American Business Law Journal* on 39 (4), pp. 683–710.